**项目报告**

**施云翀，马嘉欣，徐奕**

**2025年1月**

**1 引言**

本项目为SI100B课程的最终项目，项目文件上传至：<https://github.com/ProjectEggGame/PikyorEgg.git>。

我们的游戏是一款任务闯关类的养成游戏。玩家将作为一只被主人抛弃的小鸡醒来，通过与世界的交互，完成一系列任务，最终完成孵出小鸡、繁育后代的目标，这个游戏希望能通过沉浸式的体验让玩家感受到成长和生育的美好，从而增加我国生育率。

在主世界中，你可以通过捡拾米粒、收集树枝来完成任务，并通过转送门进入老巫婆鸡的世界接受教育，在这个过程中，要时刻提防蠢蠢狐狸的袭击，通过回答问题寻找线索，避免坏长老鸡的攻击，找到真正的老巫婆鸡。

为了实现下蛋的任务，你需要与情敌母鸡战斗，为自己的孩子找到一只帅气公鸡当爸爸，接受他的受精。

最终，你进入蛋蛋工厂，下一个蛋并孵化你的宝宝出生。

**2 项目实施**

**2.1 场景**

**2.1.1 场景数量**

我们的游戏中一共有2个场景：

1. 村庄：村庄是游戏的主世界，小鸡可以在此收集米粒和树枝，并拥有一块自留地作为家园。
2. 老巫婆鸡世界：老巫婆的世界中善恶并存。老巫婆鸡可以帮助主角小鸡接受教育，但是前提是你要躲避狐狸的追踪和袭击，寻找线索并躲避坏长老鸡。
3. 蛋蛋工厂（？）

**2.1.2 场景镜头移动**

我们的所有场景都能够实现随着主角移动而变化视角的摄像头。

在任意窗口中，玩家可以通过按空格键切换移动视角与固定视角。

**2.1.3 互动物品**

我们的游戏中有很多种互动物品，分别为米粒、树枝、鸡窝、蛋。

其中，米粒和树枝可以被捡拾，主角与物品碰撞之后，物品会被自动捡拾，收集到的物品能够帮助主角完成任务面板中的任务。

收集足够的树枝后，玩家可以在家园中通过按下H键建造一个属于自己的鸡窝。

玩家通过努力，可以下一个属于自己的蛋。

**2.2 角色**

**2.2.1 主要角色**

主要角色是由玩家控制的小鸡。玩家通过wasd四个按键操控小鸡进行移动，其中四个方向的走动配有不同的动画。小鸡可以通过传送门进入不同场景，通过碰撞实现拾取与交互。主要角色的移动通过检测按键的按下与抬起的状态来实现，当按键被按下时，角色就会向指定方向移动，按键抬起移动停止。（按键实现方法）

（详细解释）

**2.2.2 友好NPC**

游戏中存在两种友好NPC，其一是老巫婆世界中的老巫婆鸡，玩家通过在传送门上停留3秒进入老巫婆世界，通过与老巫婆的碰撞接受教育，完成第三个目标任务。

帅气公鸡是第二个友好NPC,在主世界中为主角的蛋提供一个爸爸。玩家要接受他的受精，完成第四个目标任务

**2.2.3 敌人**

游戏中共有三个敌人，分别为

第一个敌人是蠢蠢狐狸，蠢蠢狐狸是出现频率最高的敌人，他们守卫在主世界的多米区，也在老巫婆世界出现。蠢蠢狐狸具备实时战斗系统。

第二个敌人是情敌母鸡，作为与主角争夺帅气公鸡的情敌，她们的战斗力很强。

第三个敌人是坏长老鸡，坏长老鸡的攻击属性比较强，能够重伤主角，所以玩家应尽量避开。

**2.3 游戏机制**

**2.3.1 核心机制**

作为一款养成类任务闯关游戏，我们为玩家提供了一个任务面板。玩家可以按下Tab键进行查看。玩家通过操作主要角色移动来与世界中的实体碰撞，碰撞后触发相应事件。玩家在世界中生存，完成任务面板提供的任务，并最终下一个自己的蛋。

**2.3.2 碰撞系统**

**2.3.3 资源系统**

**2.4 LLM**

**2.4.1 对话系统**

**2.4.2 决策系统**

**2.5 游戏性**

**2.5.1 菜单**

我们的游戏有三种菜单，分别为游戏开始时呈现的主菜单、任务菜单和暂停菜单，游戏过程中，我们可以随时按Esc键进入esc菜单，获取需要的帮助，或者按Tab键进入任务菜单，找到需要完成的下一个任务。

**2.5.2 BGM**

我们的游戏在不同场景中播放不同的BGM。

**2.6 代码**

- assets/ 所有静态图像、声音资源

- block/ 方块相关逻辑

  - block.py 所有的方块类和子类

  - manager.py 方块资源管理器，将方块类和方块ID一一对应，减少循环import、局部import

- entity/ 实体相关逻辑

  - active\_skill.py 所有主动技能类

  - enemy.py 所有敌对生物类

  - entity.py 实体基类，及玩家、蛋等末端实体类

  - manager.py 实体资源管理器，将实体类和实体ID一一对应，减少循环import、局部import

  - skill.py 技能基类及所有被动技能类- interact/ 交互相关逻辑

  - \_\_init\_\_.py 所有玩家交互信息（因为久远的原因放在了\_\_init\_\_.py）

  - status.py 状态类，用于保存和简化处理玩家交互信息

- item/ 道具，已弃用

- LLA/ AI交互逻辑

- music/ 声音、音效

  - music.py 音像资源管理器

- render/ 渲染逻辑

  - font.py 管理所有字体资源

  - renderable.py 渲染基类，及所有可渲染对象（因为久远实际可弃用）

  - renderer.py 渲染器

  - resource.py 管理纹理图片资源

- save/ 存档逻辑

  - configs.py 处理游戏配置文件，保存玩家设置

  - save.py 处理游戏存档数据

- user/ 玩家信息。由游戏自动生成，首次运行前不存在

  - archive/ 所有存档文件

  - config.json 游戏配置文件

- utils/ 所有工具模块和类工具模块

  - \_\_init\_\_.py 日志、报错信息优化（因为久远的原因放在了\_\_init\_\_.py）

  - element.py 游戏元素基类。与Item协作，现可弃用

  - error.py 游戏内定义的错误类

  - game.py 游戏框架逻辑，游戏管理器

  - sync.py 用于防止多线程数据冲突造成游戏进行不协调

  - text.py RenderableString类，简化文本渲染流程

  - vector.py 向量类，提高位置计算相关代码可读性和流程

- window/ 窗口相关逻辑

  - hud.py HUD类，在游戏中实时现实游戏信息，包括血条、成长值、文字提示等

  - ingame.py 游戏内窗口，包括状态窗口、任务窗口等

  - input.py 输入窗口，主要包含AI助手窗口

  - widget.py 窗口按钮

  - window.py 窗口基类，及开始窗口等游戏流程外窗口；鼠标浮窗

- world/ 游戏世界（即场景）相关逻辑

  - world.py 所有世界（场景）类

- main.py 游戏入口点

- test\_code.py 用于临时测试、单元测试

**3 创意**

**3.1剧情窗口**

为了增加玩家沉浸式的体验，游戏设置了剧情窗口，通过游戏开始前的剧情介绍，让玩家沉浸式进入游戏的世界。在剧情窗口中，玩家可以通过按钮与窗口互动，逐页翻动剧情窗口或直接跳过。

**3.2 动画与特效**

**3.2.1 实体动画**

游戏主角小鸡及所有NPC都具有丰富的动画，向上下左右四个方向的移动都配有不同的行走动画，在织鸡窝、下蛋等任务完成过程中也有表示进行中的动画，让游戏更加生动。

**3.2.2 界面动画**

在所有菜单界面，将鼠标悬浮在按钮上方时，按钮底色及文字颜色会进行对应更改，增强菜单的交互性。

**3.3 技能机制**

通过吃米、收集树枝等功能，

**3.4 个性化设置窗口**

**3.5 本地保存功能**